MISSION 4: Animatronics Lesson 2 (Objectives 4-7)	**Time Frame: 35-45 minutes**
Project Goal: Students count the number of guests using a button press and display the count using line sensor LEDs. **Learning Targets** • I can increment a counter variable. • I can use buttons.was_pressed() in an if statement to control program flow. • I can display a number using line sensor LEDs. • I can break a loop. • I can add a beep to a program. • I can debounce a button press.	**Key Concepts** • Increments (and decrements) are used for updating variables like counters. • Button presses (inputs), LEDs (outputs) and speaker sounds (outputs) are part of the user interface. They allow the user to interact with the CodeBot. • A button press can "bounce," or detect an additional press that didn't happen. Using an extra buttons.was_pressed() after a short delay can debounce a button press. • A number can be displayed by using LEDs.
Assessment Opportunities • Mission 4 Lesson 2 Log (digital) • Submit completed program ***CountGuests*** • Submit the program with extensions • Mission 4 Obj. 4-7 Review Kahoot!	**Success Criteria** ☐ Increment a variable to count a button press ☐ Use a variable to turn on a line sensor LED ☐ Add a beep when a button is pressed ☐ Debounce a button press
Teacher Materials in Learning Portal • Mission 4 Lesson 2 Slides • Mission 4 Lesson 2 Log • Mission 4 Lesson 2 Answer Key	**Additional Resources** • Mission 4 Obj. 4-7 Review Kahoot! • ***CountGuests*** sample code (learning portal) • ***CountGuests_ext*** sample code (learning portal)

Vocabulary
- **Break:** Exit the nearest enclosing loop.
- **Increment:** Increase by one.
- **Debounce:** Reset the internal status of a button so the press isn't counted twice.

New Python Code

`break`	Break out of a loop
`leds.ls_num(n_guests, True)`	Turn on a line sensor LED, using a variable to indicate which LED
`n_guests = n_guests + 1`	Increment
`spkr.pitch(440)`	Play a tone on the speaker; the argument is the pitch frequency in Hertz
`spkr.off()`	Turn off the speaker (usually after a short delay)
`buttons.was_pressed(0)`	Debounce a button press by resetting the internal status (after a delay)

Real World Applications
Computers and sensors are used in many real world applications, especially to automate tasks, improve efficiency and gather data for better decision making. Some examples are:
- In manufacturing, like parts on an assembly line

- In retail, such as customers entering stores or how many items of something are on the shelf
- In agriculture, with crop monitoring and livestock counting
- In healthcare, such as counting blood cells
- In smart cities, controlling traffic flow and monitoring pedestrians

Teacher Notes:

- RECOMMENDATION: Use the slides instead of instructions in CodeSpace and CodeTrek. This mission is divided into 4 lessons that cover all the information, but chunked more. All goals will still be met, but students are asked to create a new program and only focus on the new material. The programs will be combined in Lesson 4. If you choose to use the instructions in CodeSpace and the CodeTrek, then don't use the slides.
- On Objective 5, students test the program by pressing the button and lighting up line sensor LEDs. They may find that a single press lights up more than one LED. I recommend bringing this up. Then see if the same thing happens for Obj. 6. This action is called "bouncing." Students learn about and fix this in Obj. 7.
- The quiz after Obj. 7 includes a question that may be difficult. The third question has a while loop with the condition i < 3. In this problem, the i variable is incremented outside the loop. It never changes, so the loop becomes infinite.

Extensions / Cross-Curricular:

- Use the user LEDs to display the number of guests. Go up to 8 guests before breaking the loop.
- Use the user LEDs to display the number of guests. Get a random number between 5 and 8 to break the loop, so the number of guests can be different.
- Change the tone of the beep with every button press by adding another variable.
- **MATH:** When programming, you can increment a variable with something other than 1. Incrementing by 2 can give you all the even numbers. Incrementing by other values is like skip counting. Practice skip counting, and then think about what the code would look like. What would the initial value be?
- **LANGUAGE ARTS:** Write about a real-world application that counts things using some kind of sensor or input. (See real world applications above.)
- **SCIENCE:** Learn about the mechanics of a button.
- Supports **language arts** through reading instructions, guided notes, and reflection writing.

Preparing for the lesson:

- Look through the slides and workbook. Decide what materials you want to use for presenting the lesson. The slides can be converted to Google Slides. They can be projected on a large screen. The workbook (if used) can be printed or remain digital through your LMS and given to students.
- Be familiar with the mission log assignment and the questions they will answer. Prepare the assignment to give through your LMS.
- Look over the quiz in advance, and prepare for potential questions students may have.
- If you have a word wall, or another form of vocabulary presentation, prepare the new terms.

Lesson Tips and Tricks:

💡 **Teaching tip:**

You can use a variety of discussion strategies to get the most engagement from your students. For example, you can have students write their answers before asking anyone for an answer. You can use one of many think-pair-share methods. You can have students write their answer and share with someone, and then have other students share answers they heard from their peers. You can randomly select students to answer.

👫 **Pre-Mission Warm-up:** -- slide 2

Students can write in their log first and then share, or discuss first and then write in their log. These warm-up questions review code from Mission 4 Lesson 1. Students can share their answers, or compare with each other.

- Question: What line of code increments a counter?
- Question: How can you avoid a ValueError?

💻 Mission 4 Lesson 2 Activities:

The Chrome browser works best, but other browsers also support CodeSpace. Each student will complete a Mission Log. Students could work in pairs through the lesson, or they can work individually.

💡 **Teaching tip: Mission Introduction** -- slides 3-5
This mission is divided up into four lessons. The second lesson focuses only on the second goal. These slides review the task and goal. Students answer two questions in their mission log.

💡 **Teaching tip: Objective #4** -- slides 6-9
Students start a new file. This isn't part of the instructions in CodeSpace or CodeTrek. But this way students can focus on the new concepts and not worry about a program getting too long too early. The code is given slide 9. The program starts with blinking a single LED (like objective 1) and breaking the loop.

💡 **Teaching tip: Objective #5** -- slides 10-16
Follow the slides instead of the instructions in CodeSpace or CodeTrek. The goals will still be met. Students turn on a line sensor LED and increment the counter variable before breaking the loop. Students will delete the blinking LED code and then add to the if statement. All code is given on the slides.

When students run their code, they may notice that a single button presses lights up more than one line sensor LED. This is called "bouncing," and they may also see this effect in Objective 6. You can discuss what is happening, or at least acknowledge that it is happening. Then let students know they will learn about this in Objective 7 and get the code to fix it.

💡 **Teaching tip: Objective #6** -- slides 17-19
Students add a beep to the if statement, indicating every button press. Two functions are introduced that play a tone and stop the speaker. Students need to code a delay in between the two functions.

💡 **Teaching tip: Objective #7** -- slides 20-25
Slides 20-22 discuss button bouncing. Slide 23 and 24 give the algorithm for the solution. Students just need to add one line of code to the program.

At the end of this objective, students should have a nicely working program that counts button presses and displays the number by turning on line sensor LEDs. It also beeps for each button press. When the count reaches 5, the loop breaks and the program ends.

💡 **Teaching tip: Quiz** -- slide 26
Students take a ❓ short quiz over objectives 1-7. The quiz questions are shown below.

The third question has a while loop with the condition i < 3. In this problem, the i variable is incremented outside the loop. It never changes, so the loop becomes infinite.

💡 **Teaching tip: Extensions** -- slide 27
If you have time, students can do an extension. If they do an extension, they might want to do a "File-Save As" so they have their original code, which they will use later. A code solution for all three extensions is available in the teacher resources.

Optional: 🗝️ Mission 4 Obj 4-7 Kahoot! Review.

A review Kahoot! Is available for these four objectives. You can do the Kahoot together as a class, or assign it independently.

👫 Post-Mission Reflection:

The post-mission reflection asks students to review counting guests and debouncing button presses.

- You can use an extension or cross-curricular activity as post-mission activity.
- End by collecting the Mission 4 Lesson 2 Log.

SUCCESS CRITERIA:

- ❑ Increment a variable to count a button press
- ❑ Use a variable to turn on a line sensor LED
- ❑ Add a beep when a button is pressed
- ❑ Debounce a button press

Quiz Questions (see next page):

Checkpoint

Your project is going well!

- *Animatronics* is a great way to expand your ***coding skills***.
- Now take a minute or two to review what you've learned.

+5

```
n = 7
n = n + 1
```

What is the value of `n` ***after*** the statement `n = n + 1` runs?

☒ 1 ☑ 8 ☒ 'm' ☒ 6 ☒ 7

Will the following program turn the LED on? +5

```
from botcore import leds
while False:
    leds.user_num(0, True)
```

☒ Yes. ☑ No.

+5

```
from botcore import leds
from time import sleep

i = 0
while i < 3:
    leds.user_num(0, True)
    sleep(1.0)
    leds.user_num(0, False)
    sleep(1.0)

i = i + 1
```

How many times will the LED flash when the code above runs?

☒ Three times. ☒ Two times. ☑ Infinite times. The increment is outside the loop.

The `buttons.was_pressed(0)` function returns `True` when:

- [x] The button has been pressed since was_pressed(0) was last called.
- [ ] The button has been pressed since the program started.
- [ ] The button was pressed in the last 100 milliseconds.